What is Claimed is:

1   1.   A method comprising:

2   dividing a data word into data sub-words onto sub-word paths;

3   allowing communication between the sub-word paths; and

4   encoding the data sub-words into encoded data sub-words;

5   such that an information content of the data word is spread between the encoded

6   data sub-words and the weight of the encoded data sub-words.

1   2.   A method comprising:

2   allowing communication between sub-word paths; and

3   decoding encoded data sub-words into data sub-words;

4   such that the data sub-words form a data word whereby an information

5   content of the data word is concentrated back into the data word.

1   3.   An encoder module, to encode a data word, whose data elements occupy

2   at least a first logic state and a second logic state, said encoder module

3   comprising:

4   at least two sub-word paths, each of said at least two sub-word paths to

5   receive a data sub-word, comprising a set of data elements of the

6   data word; and

7   an encoder coupled with said at least two sub-word paths, said encoder to

8   encode the data sub-word into an encoded data sub-word;

9   such that encoded data sub-words form an encoded data word wherein an

10   information content of the data word is spread between the encoded data sub-

11   words and the weight of the encoded data sub-words.

1   4.   An encoder module, as in claim 3, wherein said encoder module allows

2   communication of a group of data elements from the data word, such that

3   information is shared between said at least two sub-word paths.

1   5.   An encoder module, as in claim 4, wherein the information is a data sub-

2   word weight.

1   6.   An encoder module, as in claim 4, wherein the information is at least one

2   data element from the data sub-word.

1   7.   An encoder module, as in claim 4, wherein the information is at least one

2   data element from the data word.

1   8.   An encoder as recited in claim 4, wherein said encoder inverts all the data

2   elements, within the data sub-word, whose significance is less than or equal to

3   the significance of a particular data element within the data sub-word.

1   9.   An encoder as recited in claim 4, wherein said encoder inverts all encoded

2   data elements, within the encoded data sub-word, whose significance is less than

3   or equal to the significance of a particular encoded data element within the

4   encoded data sub-word.

1   10.   An encoder as recited in claim 4, wherein the number of elements in the

2   encoded data sub-word is at least one greater than the number of elements in the

3   data sub-word.

CLAIMS

1 11.   An encoder module, as in claim 4, further comprising encoding logic for
2 encoding values represented by the data words that are not encoded by said
3 encoder.

1 12.   An encoder module, as in claim 4, further comprising a code governor,
2 wherein said code governor sets a logic state of at least one particular data
3 element location in the encoded data word such that the number of data
4 elements in each logic state is constant for each encoded data word.

1 13.   A code governor, as recited in claim 12, wherein said code governor, uses
2 in part, the information shared between said at least two sub-word paths to set
3 the logic state of at least one particular data element in the encoded data word.

1 14.   An encoder module, as in claim 12, further comprising a parity line to
2 change the weight of the encoded data word by adding an element to the
3 encoded data word.

1 15.   A parity line, as recited in claim 14, wherein said parity line, uses in part,
2 the information shared between said at least two sub-word paths to set the logic
3 state of the element output onto said parity line.

1 16.   An encoder as recited in claim 4, wherein said encoder encodes the data
2 word by associating the data word with an encoded data word which is selected
3 from a set of encoded data words where the encoded data word is of length z
4 elements with a fixed number of data elements in each logic state.

CLAIMS

1   17.   An encoder as recited in claim 4, wherein said encoder encodes the data

2   sub-word by associating the data sub-word with the encoded data sub-word in a

3   user defined way.

1   18.   An encoder as recited in claim 4, wherein said encoder is a binomial

2   encoder.

1   19.   An encoder module as recited in claim 4, wherein said encoder module is

2   disposed on an integrated circuit die.

1   20.   An encoder module, as in claim 4, further comprising a Vdd power rail

2   wherein a sum of the current flowing in said Vdd power rail is substantially

3   constant.

1   21.   An encoder module, as in claim 4, further comprising:

2   a code governor; and

3   a Vdd power rail;

4   wherein said code governor sets the logic state of at least one particular data

5   element location in the encoded data word, such that a sum of the current

6   flowing in said Vdd power rail is substantially constant.

1   22.   A code governor, as recited in claim 21, wherein said code governor, uses

2   in part, the information shared between said at least two sub-word paths to set a

3   logic state of at least one particular data element in the encoded data word.

CLAIMS

1 23.   An encoder module, as in claim 21, further comprising a parity line to

2 change the weight of the encoded data word by adding an element to the

3 encoded data word.

1 24.   A parity line, as recited in claim 23, wherein the information shared

2 between the data sub-words is used, in part, to set a logic state of the element

3 output onto said parity line.

1 25.   An encoder module, as in claim 4, further comprising a *Vss* power rail

2 wherein a sum of a current flowing in said *Vss* power rail is substantially

3 constant.

1 26.   An encoder module, as in claim 4, further comprising:

2   a code governor; and

3   a *Vss* power rail;

4 wherein said code governor sets the logic state of at least one particular data

5   element location in the encoded data word, such that a sum of the current

6   flowing in said *Vss* power rail is substantially constant.

1 27.   A code governor, as recited in claim 26, wherein said code governor, uses

2 in part, the information shared between said at least two sub-word paths to set a

3 logic state of at least one particular data element in the encoded data word.

1 28.   An encoder module, as in claim 26, further comprising a parity line to

2 change the weight of the encoded data word by adding an element to the

3 encoded data word.

CLAIMS

1 29. A parity line, as recited in claim 28, wherein the information shared
2 between the data sub-words is used, in part, to set a logic state of the element
3 output onto said parity line.

1 30. An encoder module, as in claim 4, further comprising encoded data lines
2 wherein a sum of the current flowing, in said encoded data lines, is substantially
3 constant.

1 31. An encoder module, as in claim 4, further comprising:
2 a code governor, wherein said code governor sets the logic state of at least one
3 particular data element location in the encoded data word; and
4 encoded data lines, connected with said encoder;
5 such that a sum of the current flowing in said encoded data lines, connected with
6 said encoder is substantially constant.

1 32. A code governor, as recited in claim 31, wherein said code governor, uses
2 in part, the information shared between said at least two sub-word paths to set a
3 logic state of at least one particular data element in the encoded data word.

1 33. An encoder module, as in claim 31, further comprising a parity line to
2 change the weight of the encoded data word by adding an element to the
3 encoded data word.

1 34. A parity line, as recited in claim 33, wherein the information shared
2 between the data sub-words is used, in part, to set a logic state of the element
3 output onto said parity line.

CLAIMS

1   35.   A method of encoding a data word, whose data elements occupy at least a

2   first logic state and a second logic state, said method comprising:

3   receiving data sub-words onto sub-word paths, the data sub-words

4   comprising sets of data elements of the data word; and

5   encoding the data sub-words into encoded data sub-words;

6   such that the encoded data sub-words form an encoded data word wherein an

7   information content of the data word is spread between the encoded data

8   sub-words and the weight of the encoded data sub-words.

1   36.   A method, as in claim 35, further comprising allowing communication of a

2   group of data elements from the data word, between the sub-word paths, such

3   that information may be shared between the sub-word paths.

1   37.   A method, as in claim 36, wherein the information in said allowing

2   communication is a data sub-word weight.

1   38.   A method, as in claim 36, wherein the information in said allowing

2   communication is at least one data element from the data sub-word.

1   39.   A method, as in claim 36, wherein the information is at least one data

2   element from a data word.

1   40.   A method, as in claim 36, wherein said method further comprises

2   inverting all data elements, within a data sub-word, whose significance is less

3   than or equal to the significance of a particular data element within the data sub-

4   word.

1　41.　A method, as in claim 36, wherein said method further comprises

2　inverting all encoded data elements, within an encoded data sub-word, whose

3　significance is less than or equal to the significance of a particular encoded data

4　element within the encoded data sub-word.

1　42.　A method, as in claim 36, wherein said encoding encodes a data word by

2　associating the data word with an encoded data word which is selected from a

3　set of encoded data words where the encoded data word is of length z elements

4　with a fixed number of data elements in each logic state.

1　43.　A method, as in claim 36, wherein said encoding encodes a data sub-word

2　by associating the data sub-word with the encoded data sub-word in a user

3　defined way.

1　44.　A method, as in claim 36, wherein the encoded data word is balanced.

1　45.　A method, as in claim 36, wherein the encoded data word is substantially

2　balanced.

1　46.　A method, as in claim 36, further comprising using a code governor to

2　change the weight of the encoded data word.

1　47.　A method, as in claim 36, wherein said using a code governor is based at

2　least in part on said allowing communication.

1　48.　A method, as in claim 47, further comprising using a parity line to change

2　the weight of the encoded data word.

1   49.   A method, as in claim 48, wherein said using a parity line is based at least

2   in part on said allowing communication.

1   50.   A method, as in claim 36, wherein said encoding is a means for binomially

2   encoding the data word.

1   51.   A method, as in claim 50, further comprising adding 1 to the largest

2   decimal number to be encoded.

1   52.   A method, as in claim 50, wherein the encoded data word is balanced.

1   53.   A method, as in claim 50, wherein the encoded data word is substantially

2   balanced.

1   54.   A method, as in claim 50, further comprising using a code governor to

2   change the weight of the encoded data word.

1   55.   A method, as in claim 54, wherein said using a code governor is based at

2   least in part on said allowing communication.

1   56.   A method, as in claim 54, further comprising using a parity line to change

2   the weight of the encoded data word.

1   57.   A method, as in claim 56, wherein said using a parity line is based at least

2   in part on said allowing communication.

1 58. A decoder module to decode an encoded data word, whose encoded data

2 elements occupy at least a first logic state and a second logic state, said decoder

3 module comprising:

4   at least two sub-word paths, each of said at least two sub-word paths to

5   receive an encoded data sub-word, comprising a set of data

6   elements of the encoded data word and to allow communication,

7   between said at least two sub-word paths, such that information is

8   shared between said at least two sub-word paths; and

9   a decoder coupled with said at least two sub-word paths, said decoder to

10   decode the encoded data sub-word into a data sub-word;

11   such that data sub-words form a data word.

1 59. A decoder as recited in claim 58, wherein said decoder inverts all data

2 elements, within the data sub-word, whose significance is less than or equal to

3 the significance of a particular data element within the data sub-word.

1 60. A decoder as recited in claim 58, wherein said decoder inverts all encoded

2 data elements, within the encoded data sub-word, whose significance is less than

3 or equal to the significance of a particular encoded data element within the

4 encoded data sub-word.

1 61. A decoder as recited in claim 58, wherein the number of elements in the

2 data sub-word is at least one less than the number of elements in the encoded

3 data sub-word.

CLAIMS

1 62.   A decoder module, as in claim 58, further comprising decoding logic for
2 decoding a value represented by the encoded data word that is not decoded by
3 said decoder.

1 63.   A decoder as recited in claim 58, wherein said decoder decodes the
2 encoded data sub-word by associating the encoded data sub-word with the data
3 sub-word in a user-defined way.

1 64.   A decoder as recited in claim 58, wherein said decoder is a binomial
2 decoder.

1 65.   A decoder module, as in claim 58, wherein said decoder module is
2 disposed on an integrated circuit die.

1 66.   A method of decoding an encoded data word, whose encoded data
2 elements occupy at least a first logic state and a second logic state, said method
3 comprising:
4   receiving encoded data sub-words onto sub-word paths, the encoded data
5   sub-words comprising sets of data elements of the encoded data
6   word;
7   allowing communication between the sub-word paths, such that
8   information may be shared between the sub-word paths; and
9   decoding the encoded data sub-words into data sub-words;
10 such that the data sub-words form a data word.

1 67.   A method, as in claim 66, wherein the information, in said allowing
2 communication, is an encoded data sub-word weight.

1   68.   A method, as in claim 66, wherein the information, in said allowing

2   communication, is at least one encoded data element from an encoded data sub-

3   word.

1   69.   A method, as in claim 66, wherein the information, in said allowing

2   communication, is at least one encoded data element from the encoded data

3   word.

1   70.   A method, as in claim 66, wherein said method further comprises

2   inverting all encoded data elements, within an encoded data sub-word, whose

3   significance is less than or equal to the significance of a particular encoded data

4   element within the encoded data sub-word.

1   71.   A method, as in claim 66, wherein said method further comprises

2   inverting all encoded data elements, within an encoded data sub-word, whose

3   significance is less than or equal to the significance of a particular encoded data

4   element within the encoded data sub-word.

1   72.   A method, as in claim 66, wherein said decoding, decodes an encoded

2   data sub-word by associating the encoded data sub-word with a data sub-word

3   in a user defined way.

1   73.   A method, as in claim 66, wherein said decoding is a means for binomially

2   decoding a data sub-word.

1   74.   A method, as in claim 73, further comprising adding 1 to a decimal value

2   to be decoded.

CLAIMS

1   75.   A data processing system comprising:

2   at least two sub-word paths, each of said at least two sub-word paths to

3   receive a data sub-word, comprising a set of data elements of a data

4   word;

5   an encoder coupled with said at least two sub-word paths, said encoder to

6   encode the data sub-word into an encoded data sub-word; such

7   that encoded data sub-words form an encoded data word wherein

8   an information content of the data word is spread between the

9   encoded data sub-words and the weight of the encoded data sub-

10   words.

11   a parallel encoded data line bus coupled with said at least two sub-word

12   paths to receive the encoded data sub-words and to facilitate

13   transmission of the encoded data sub-words;

14   at least two sub-word paths coupled with said parallel encoded data line

15   bus, each of said at least two sub-word paths to receive an encoded

16   data sub-word, comprising a set of data elements of the encoded

17   data word and allowing communication, between said at least two

18   sub-word paths, such that information is shared between said at

19   least two sub-word paths; and

20   a decoder coupled with said at least two sub-word paths, said decoder to

21   decode the encoded data sub-word into the data sub-word;

22   such that data sub-words form the data word.

1   76.   A data processing system, as in claim 75, wherein said data processing

2   system allows communication of a group of data elements from the data word,

3   such that information is shared between said at least two sub-word paths.

CLAIMS

1 77. A data processing system, as in claim 76, wherein the information is a data

2 sub-word weight.

1 78. A data processing system, as in claim 76, wherein the information is at

2 least one data element from the data sub-word.

1 79. A data processing system, as in claim 76, wherein the information is at

2 least one data element from the data word.

1 80. An encoder as recited in claim 76, wherein said encoder inverts all the

2 data elements, within the data sub-word, whose significance is less than or equal

3 to the significance of a particular data element within the data sub-word.

1 81. An encoder as recited in claim 76, wherein said encoder inverts all

2 encoded data elements, within the encoded data sub-word, whose significance is

3 less than or equal to the significance of a particular encoded data element within

4 the encoded data sub-word.

1 82. An encoder as recited in claim 76, wherein the number of elements in the

2 encoded data sub-word is at least one greater than the number of elements in the

3 data sub-word.

1 83. A data processing system, as in claim 76, further comprising encoding

2 logic for encoding values represented by the data words that are not encoded by

3 said encoder.

CLAIMS

1 84.   A data processing system, as in claim 76, further comprising a code

2 governor, wherein said code governor sets a logic state of at least one particular

3 data element location in the encoded data word such that the number of data

4 elements in each logic state is constant for each encoded data word.

5

1 85.   A code governor, as recited in claim 84, wherein said code governor, uses

2 in part, the information shared between said at least two sub-word paths to set

3 the logic state of at least one particular data element in the encoded data word.

1 86.   A data processing system, as in claim 84, further comprising a parity line

2 to change the weight of the encoded data word by adding an element to the

3 encoded data word.

1 87.   A parity line, as recited in claim 86, wherein said parity line, uses in part,

2 the information shared between said at least two sub-word paths to set the logic

3 state of the element output onto said parity line.

1 88.   An encoder as recited in claim 76, wherein said encoder encodes the data

2 word by associating the data word with an encoded data word which is selected

3 from a set of encoded data words where the encoded data word is of length z

4 elements with a fixed number of data elements in each logic state.

1 89.   An encoder as recited in claim 76, wherein said encoder encodes the data

2 sub-word by associating the data sub-word with the encoded data sub-word in a

3 user defined way.

CLAIMS

1 90. An encoder as recited in claim 76, wherein said encoder is a binomial

2 encoder.

1 91. A data processing system as recited in claim 76, wherein said encoder

2 module is disposed on an integrated circuit die.

1 92. A data processing system, as in claim 76, further comprising a *Vdd* power

2 rail wherein a sum of the current flowing in said *Vdd* power rail is substantially

3 constant.

1 93. A data processing system, as in claim 76, further comprising:

2   a code governor; and

3   a *Vdd* power rail;

4 wherein said code governor sets the logic state of at least one particular data

5   element location in the encoded data word, such that a sum of the current

6   flowing in said *Vdd* power rail, is substantially constant.

1 94. A code governor, as recited in claim 93, wherein said code governor, uses

2 in part, the information shared between said at least two sub-word paths to set a

3 logic state of at least one particular data element in the encoded data word.

1 95. A data processing system, as in claim 93, further comprising a parity line

2 to change the weight of the encoded data word by adding an element to the

3 encoded data word.

CLAIMS

1   96.   A parity line, as recited in claim 95, wherein the information shared

2   between the data sub-words is used, in part, to set a logic state of the element

3   output onto said parity line.

1

1   97.   A data processing system, as in claim 76, further comprising a *Vss* power

2   rail wherein a sum of a current flowing in said *Vss* power rail is substantially

3   constant.

1   98.   A data processing system, as in claim 76, further comprising:

2   a code governor; and

3   a *Vss* power rail;

4   wherein said code governor sets the logic state of at least one particular data

5   element location in the encoded data word, such that a sum of the current

6   flowing in said *Vss* power rail is substantially constant.

1   99.   A code governor, as recited in claim 98, wherein said code governor, uses

2   in part, the information shared between said at least two sub-word paths to set a

3   logic state of at least one particular data element in the encoded data word.]

1   100.   A data processing system, as in claim 98, further comprising a parity line

2   to change the weight of the encoded data word by adding an element to the

3   encoded data word.

1   101.   A parity line, as recited in claim 100, wherein the information shared

2   between the data sub-words is used, in part, to set a logic state of the element

3   output onto said parity line.

1   102.   A data processing system, as in claim 76, further comprising encoded data

2   lines wherein a sum of the current flowing in said parallel encoded data line bus

3   is substantially constant.

1   103.   A data processing system, as in claim 76, further comprising:

2   a code governor, wherein said code governor sets the logic state of at least one

3   particular data element location in the encoded data word;

4   such that a sum of the current flowing in said parallel encoded data line bus is

5   substantially constant.

1   104.   A code governor, as recited in claim 103, wherein said code governor, uses

2   in part, the information shared between said at least two sub-word paths to set a

3   logic state of at least one particular data element in the encoded data word.

1   105.   A data processing system, as in claim 103, further comprising a parity line

2   to change the weight of the encoded data word by adding an element to the

3   encoded data word.

1   106.   A parity line, as recited in claim 105, wherein the information shared

2   between the data sub-words is used, in part, to set a logic state of the element

3   output onto said parity line.

1   107.   A decoder as recited in claim 76, wherein said decoder inverts all data

2   elements, within the data sub-word, whose significance is less than or equal to

3   the significance of a particular data element within the data sub-word.

CLAIMS

1 108. A decoder as recited in claim 76, wherein said decoder inverts all encoded

2 data elements, within the encoded data sub-word, whose significance is less than

3 or equal to the significance of a particular encoded data element within the

4 encoded data sub-word.

1 109. A decoder as recited in claim 76, wherein the number of elements in the

2 data sub-word is at least one less than the number of elements in the encoded

3 data sub-word.

1 110. A data processing system, as in claim 76, further comprising decoding

2 logic for decoding a value represented by the encoded data word that is not

3 decoded by said decoder.

1 111. A decoder as recited in claim 76, wherein said decoder decodes the

2 encoded data sub-word by associating the encoded data sub-word with the data

3 sub-word in a user-defined way.

1 112. A decoder as recited in claim 76, wherein said decoder is a binomial

2 decoder.

1 113. A data processing system, as in claim 76, wherein said data processing

2 system is disposed on an integrated circuit die.

1 114. A method for transmitting a data word in a data processing system, said

2 method comprising:

3   receiving data sub-words onto sub-word paths, the data sub-words

4   comprising sets of data elements of the data word;

5   encoding the data sub-words into encoded data sub-words, such that the

6   encoded data sub-words form an encoded data word wherein an

7   information content of the data word is spread between the

8   encoded data sub-words and the weight of the encoded data sub-

9   words;

10   transmitting the encoded data sub-words over a parallel encoded data line

11   bus;

12   receiving the encoded data sub-words onto the sub-word paths, the

13   encoded data sub-words comprising sets of data elements of the

14   encoded data word;

15   allowing communication, between the sub-word paths, such that

16   information may be shared between the sub-word paths; and

17   decoding the encoded data sub-words into the data sub-words;

18   such that the data sub-words form the data word.

1   115   A method, as in claim 114, further comprising allowing communication of

2   a group of data elements from the data word, between the sub-word paths, such

3   that information may be shared between the sub-word paths.

1   116.   A method, as in claim 115, wherein the information in said allowing

2   communication is a data sub-word weight.

1   117.   A method, as in claim 115, wherein the information in said allowing

2   communication is at least one data element from the data sub-word.

1   118.   A method, as in claim 115, wherein the information is at least one data

2   element from a data word.

1 119.   A method, as in claim 115, wherein said method further comprises

2 inverting all data elements, within a data sub-word, whose significance is less

3 than or equal to the significance of a particular data element within the data sub-

4 word.

1 120.   A method, as in claim 115, wherein said method further comprises

2 inverting all encoded data elements, within an encoded data sub-word, whose

3 significance is less than or equal to the significance of a particular encoded data

4 element within the encoded data sub-word.

1 121.   A method, as in claim 115, wherein said encoding encodes a data word by

2 associating the data word with an encoded data word which is selected from a

3 set of encoded data words where the encoded data word is of length z elements

4 with a fixed number of data elements in each logic state.

1 122.   A method, as in claim 115, wherein said encoding encodes a data sub-

2 word by associating the data sub-word with the encoded data sub-word in a user

3 defined way.

1 123.   A method, as in claim 115, wherein the encoded data word is balanced.

1 124.   A method, as in claim 115, wherein the encoded data word is substantially

2 balanced.

1 125.   A method, as in claim 115, further comprising using a code governor to

2 change the weight of the encoded data word.

1   126.   A method, as in claim 125, wherein said using a code governor is based at

2   least in part on said allowing communication.

1   127.   A method, as in claim 125, further comprising using a parity line to

2   change the weight of the encoded data word.

1   128.   A method, as in claim 127, wherein said using a parity line is based at

2   least in part on said allowing communication.

1   129.   A method, as in claim 115, wherein said encoding is a means for

2   binomially encoding the data word.

1   130.   A method, as in claim 129, further comprising adding 1 to the largest

2   decimal number to be encoded

1   131.   A method, as in claim 129, wherein the encoded data word is balanced.

1   132.   A method, as in claim 129, wherein the encoded data word is substantially

2   balanced

1   133.   A method, as in claim 129, further comprising using a code governor to

2   change the weight of the encoded data word.

1   134.   A method, as in claim 133, wherein said using a code governor is based at

2   least in part on said allowing communication.

1 135. A method, as in claim 133, further comprising using a parity line to

2 change the weight of the encoded data word.

1 136. A method, as in claim 135, wherein said using a parity line is based at

2 least in part on said allowing communication.

1 137. A method, as in claim 115, wherein the information, in said allowing

2 communication, is an encoded data sub-word weight.

1 138. A method, as in claim 115, wherein the information, in said allowing

2 communication, is at least one encoded data element from an encoded data sub-

3 word.

1 139. A method, as in claim 115, wherein the information, in said allowing

2 communication, is at least one encoded data element from the encoded data

3 word.

1 140. A method, as in claim 115, wherein said method further comprises

2 inverting all encoded data elements, within an encoded data sub-word, whose

3 significance is less than or equal to the significance of a particular encoded data

4 element within the encoded data sub-word.

1 141. A method, as in claim 115, wherein said method further comprises

2 inverting all encoded data elements, within an encoded data sub-word, whose

3 significance is less than or equal to the significance of a particular encoded data

4 element within the encoded data sub-word.

CLAIMS

1   142.   A method, as in claim 115, wherein said decoding, decodes an encoded
2   data sub-word by associating the encoded data sub-word with a data sub-word
3   in a user defined way.

1   143.   A method, as in claim 115, wherein said decoding is a means for
2   binomially decoding a data sub-word.